EFFICIENT DATA COMPRESSION BY GROUPING SIMILAR DATA WITHIN A DATA SEGMENT

BACKGROUND

[0001] Storage systems, such as enterprise storage systems, may include a centralized or de-centralized repository for data that provides common data management, data protection, and data sharing functions, for example, through connections to computer systems.

BRIEF DESCRIPTION OF DRAWINGS

[0002] FIG. **1**A illustrates a first example system for data storage in accordance with some implementations.

[0003] FIG. **1**B illustrates a second example system for data storage in accordance with some implementations.

[0004] FIG. **1**C illustrates a third example system for data storage in accordance with some implementations.

[0005] FIG. **1**D illustrates a fourth example system for data storage in accordance with some implementations.

[0006] FIG. **2**A is a perspective view of a storage cluster with multiple storage nodes and internal storage coupled to each storage node to provide network attached storage, in accordance with some embodiments.

[0007] FIG. **2**B is a block diagram showing an interconnect switch coupling multiple storage nodes in accordance with some embodiments.

[0008] FIG. **2**C is a multiple level block diagram, showing contents of a storage node and contents of one of the non-volatile solid state storage units in accordance with some embodiments.

[0009] FIG. **2**D shows a storage server environment, which uses embodiments of the storage nodes and storage units of some previous figures in accordance with some embodiments.

[0010] FIG. **2**E is a blade hardware block diagram, showing a control plane, compute and storage planes, and authorities interacting with underlying physical resources, in accordance with some embodiments.

[0011] FIG. **2**F depicts elasticity software layers in blades of a storage cluster, in accordance with some embodiments.

[0012] FIG. **2**G depicts authorities and storage resources in blades of a storage cluster, in accordance with some embodiments.

[0013] FIG. **3**A sets forth a diagram of a storage system that is coupled for data communications with a cloud services provider in accordance with some embodiments of the present disclosure.

[0014] FIG. **3**B sets forth a diagram of a storage system in accordance with some embodiments of the present disclosure.

[0015] FIG. **3**C sets forth an example of a cloud-based storage system in accordance with some embodiments of the present disclosure.

[0016] FIG. **3**D illustrates an exemplary computing device that may be specifically configured to perform one or more of the processes described herein.

[0017] FIG. **4**A is an illustration of an example of storing data blocks to data segments based on languages associated with the data blocks in accordance with embodiments of the disclosure.

[0018] FIG. **4**B is an illustration of an example of a storage controller assigning data blocks to data segments based on an associated property in accordance with embodiments of the disclosure.

[0019] FIG. **4**C is an illustration of an example of storing data blocks within data segments in accordance with embodiments of the present disclosure.

[0020] FIG. **5** is an example method to store data blocks having similar properties within a data segment in accordance with embodiments of the disclosure.

[0021] FIG. **6** is an example method to allocate received data blocks to data segments based on associated properties in accordance with embodiments of the disclosure.

DETAILED DESCRIPTION

[0022] In embodiments, a storage system compresses data to reduce the number of bits needed to represent data stored at the storage system. Compressing the data saves storage capacity of the storage system. One example of data compression is lossless compression, in which none of the data being compressed is lost. In lossless compression, the number of bits is reduced by identifying and eliminating statistical redundancy in the data.

[0023] To perform compression of the data, processing logic of a storage controller of the storage system may read a string of bits from blocks of data stored at the storage system. While reading the string of bits from the blocks of data, the processing logic identifies redundancies in the data for compression. The efficiency of the compression of the data is dependent upon the proximity of the redundant data within the string of bits. For example, a compression operation performed having redundant data located within a close proximity to one another in the string of bits is more efficient than a compression operation having redundant data located farther apart in the string of bits.

[0024] In a conventional storage system, data may be programmed to a storage device of a storage system as part of a data segment along with other data to be stored at the storage system. The processing logic of the storage system may perform a compression operation on the data and other data of the data segment to identify redundancies within the data segment. However, there may be scenarios where the data and the other data within the data segment may be dissimilar types of data. For example, the data may include mostly English text while the other data may include mostly x86 code. Accordingly, because the data and other data within the data segment are dissimilar types, there may be less redundancies between the data and the other data, reducing the efficiency of the compression operation.

[0025] Turning now to embodiments of a storage system that stores data of having similar properties in the same data segment to improve the efficiency of compression operations performed by the storage system. Processing logic of a storage controller of the storage system may identify data blocks stored at one or more storage devices of the storage system. The processing logic may then read a portion of the data stored at each of the data blocks to determine one or more properties associated with each of the data blocks. The portion of the data may serve as an indicator of the properties of the underlying data stored at each of the data blocks. In embodiments, the portion of the data may correspond to one or more top bits of the data stored at the data block. For example, the processing logic may read the top three bits of data for each of the data blocks. If the top three bits of data